PathAddExtension

Return value buffer must be large enough to store returned path

Sean Barnum, Cigital, Inc. [vita[1]]

2007-04-02

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 5178 bytes

Attack Category	• Path spoofing or confusion problem • Malicious Input
Vulnerability Category	• Buffer Overflow • Unconditional
Software Context	• Filename Management
Location	• shlwapi.h
Description	The in/out buffer of the function PathAddExtension() must be large enough to hold the returned value. The PathAddExtension() routine adds an arbitrarily specified file extension to a file if and only if the file doesn't already have an extension. Does nothing to a NULL string. Will probably crash on a NULL pointer. The first parameter, pszPath, must be at least MAX_PATH characters in length to ensure that it is large enough to hold the returned string.
APIs	**Function Name** / **Comments** PathAddExtension / stores args 0+1 >> argo PathAddExtensionA / ASCII PathAddExtensionW / Unicode ATLPath::AddExtension / Overloaded wrapper of PathExtension
Method of Attack	Attacker can cause a buffer overflow if the path variable is not long enough to hold the variable. Since the extension can be of arbitrary length, this is most dangerous when the extension is provided by user input. In other situations, it may be possible for the attacker to fill the source buffer to a limit, then allow the 3+ character fixed entension to over run the buffer, possibly connecting it to other data.
Exception Criteria	

1. http://buildsecurityin.us-cert.gov/bsi-rules/35-BSI.html (Barnum, Sean)

Solutions

Solution Applicability	Solution Description	Solution Efficacy
	Ensure that destination buffer is at least MAX_PATH characters in length.	Effective.

Signature Details

BOOL PathAddExtension(
LPTSTR pszPath,
LPCTSTR pszExtension
);

Examples of Incorrect Code

```
int main(int argc, char *argv[]) {
...
TCHAR buffer_1[9] =
TEXT("file"); // Buffer may be too
small
LPSTR lpStr1;
lpStr1 = buffer_1;
TCHAR F_Ext[] = TEXT(argv[1]);
LPCTSTR lpStr3;
lpStr3 = F_Ext;
int ret_1 =
PathAddExtension(lpStr1, lpStr3);
...
}
```

Examples of Corrected Code

```
int main(int argc, char *argv[]) {
...
TCHAR buffer_1[MAX_PATH] =
TEXT("file");
LPSTR lpStr1;
lpStr1 = buffer_1;
TCHAR F_Ext[] = TEXT(argv[1]);
LPCTSTR lpStr3;
lpStr3 = F_Ext;
int ret_1 =
PathAddExtension(lpStr1, lpStr3);
...
}
```

Source Reference

- http://msdn.microsoft.com/library/default.asp?url=/library/en-us/shellcc/platform/shell/reference/shlwapi/path/pathaddextension.asp[2]

Recommended Resource

Discriminant Set

Operating System	• Windows
Languages	• C • C++

Cigital, Inc. Copyright

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

1. mailto:copyright@cigital.com